武汉纺织大学

Web应用开发课程设计

**Highcharts视图的课程设计**

**学 院： 数学与计算机学院**

**班 级： 物联网11803**

**姓 名： 李乾钰**

**学 号： 1803280324**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月15日**

目 录

[1 需求分析 1](#_Toc21282)

[1.1生成、刷新条形图(按商品数量从大到小排列) 1](#_Toc21359)

[1.2生成、刷新折线图 1](#_Toc22285)

[1.3生成、刷新饼状图 1](#_Toc1428)

[2 系统设计 1](#_Toc14603)

[2.1用户用例图 1](#_Toc24559)

[2.2 UML类图（Class Diagram） 2](#_Toc27670)

[3 系统实现 3](#_Toc14102)

[3.1 项目结构 3](#_Toc29160)

[3.2 配置文件 4](#_Toc28528)

[3.3 VO类 4](#_Toc10390)

[3.3.1 Price.java 4](#_Toc29911)

[3.3.2 Number.java 5](#_Toc16934)

[3.3.3 BrowseInfo.java 5](#_Toc14451)

[3.4 DAO类 6](#_Toc7568)

[3.4.1 PriceDao.java 6](#_Toc11801)

[3.4.2 NumberDao.java 8](#_Toc13581)

[3.4.3 BrowseInfoDAO.java 9](#_Toc19957)

[3.6 工具包 JdbcUtil.java 10](#_Toc13220)

[3.7 服务层 11](#_Toc1105)

[3.7.1 CreateColumnChartController 11](#_Toc5200)

[3.7.2 CreateLineChartController 12](#_Toc2221)

[3.7.3 CreatePieChartController 12](#_Toc32573)

[3.8 用户界面 13](#_Toc7816)

[3.8.1 index.html 13](#_Toc32226)

[3.8.2 main.css 14](#_Toc5519)

[3.8.3 jquery-3.5.1.min.js 14](#_Toc9483)

[3.8.4 highchart.js 14](#_Toc19457)

[4 系统测试 15](#_Toc3084)

[4.1 生成、刷新条形图(按商品数量从大到小排列) 15](#_Toc19860)

[4.2 生成、刷新折线图 16](#_Toc25268)

[4.3 生成、刷新饼图 17](#_Toc14876)

[5 系统总结 18](#_Toc17341)

# 1 需求分析

利用Highcharts组件，生成商品和商品数量的条形图，生成商品和销售额的折线图，生成浏览商品的占比的饼状图。数据来自于Highcharts数据库。

## 1.1生成、刷新条形图**(按商品数量从大到小排列)**

当程序运行时，在网页上生成一个条形图，数据来自本地数据库；

图表右上方显示一个按钮，点击按钮，更新数据库数据，刷新条形图。

## 1.2生成、刷新折线图

当程序运行时，在网页上生成一个条形图，数据来自本地数据库；

图表右上方显示一个按钮，点击按钮，更新数据库数据，刷新折线图。

## 1.3生成、刷新饼状图

当程序运行时，在网页上生成一个饼图，数据来自本地数据库；

图表右上方显示一个按钮，点击按钮，更新数据库数据，刷新饼状图。

# **2 系统设计**

## 2.1用户用例图

系统分为访问者，管理员两个角色，各角色的用户用例如下图2-1所示：

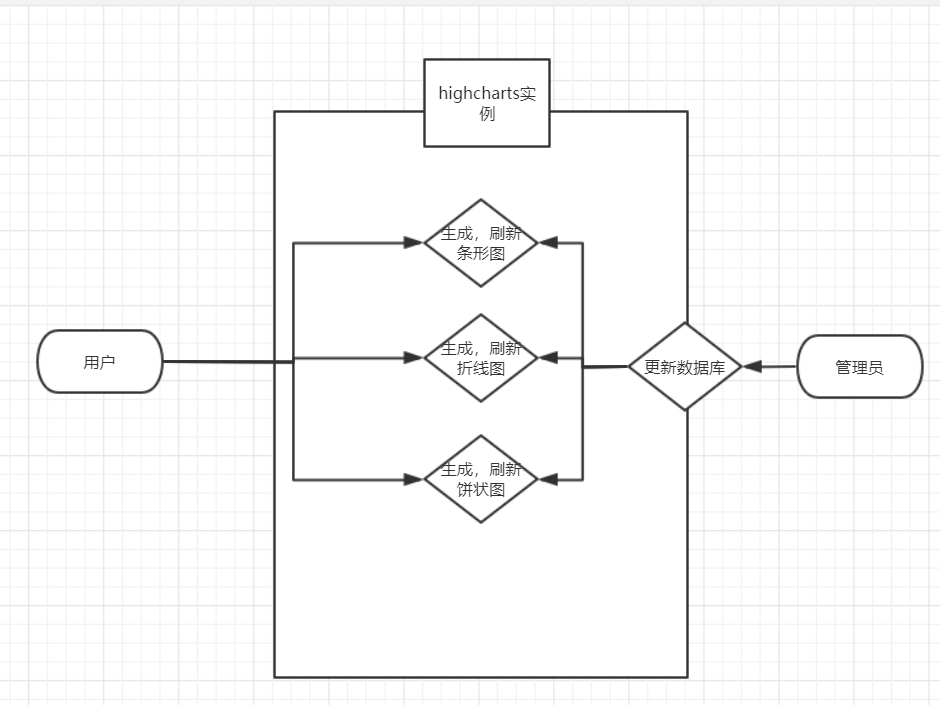


图2-1 用户用例图

## 2.2 UML类图（Class Diagram）

各类的结构及类之间的关系如图2-2-1、2-2-2、2-2-3所示：

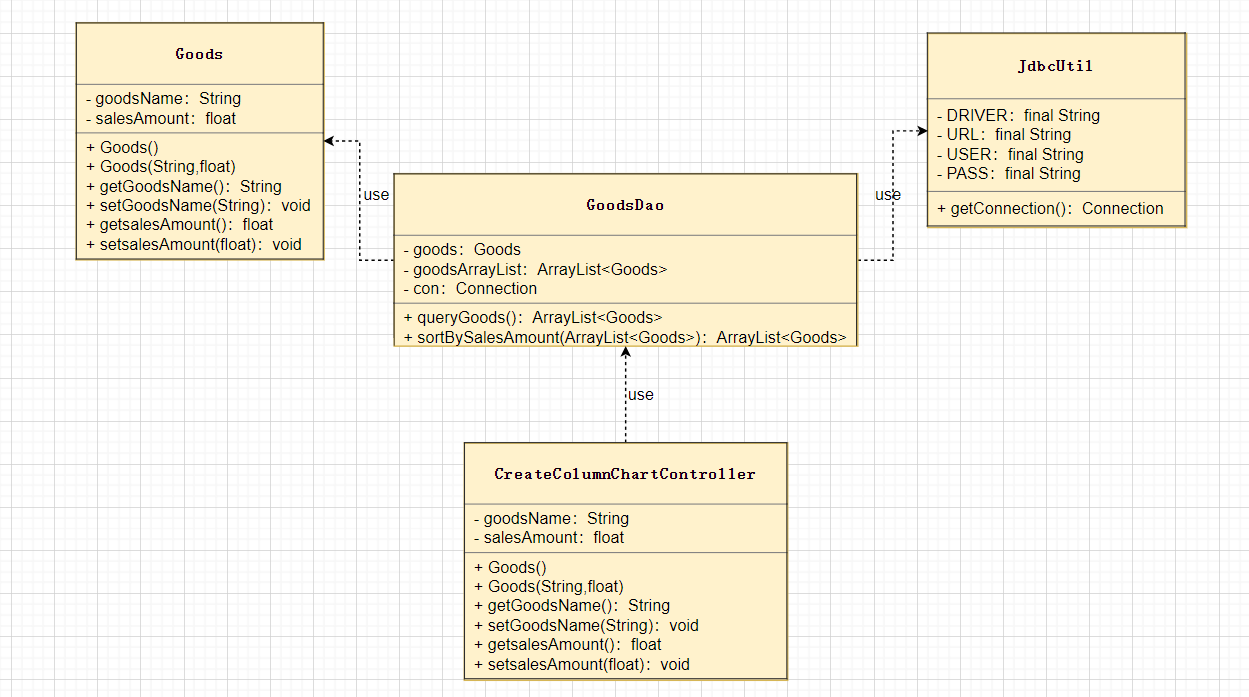


图2-2-1 功能1类图

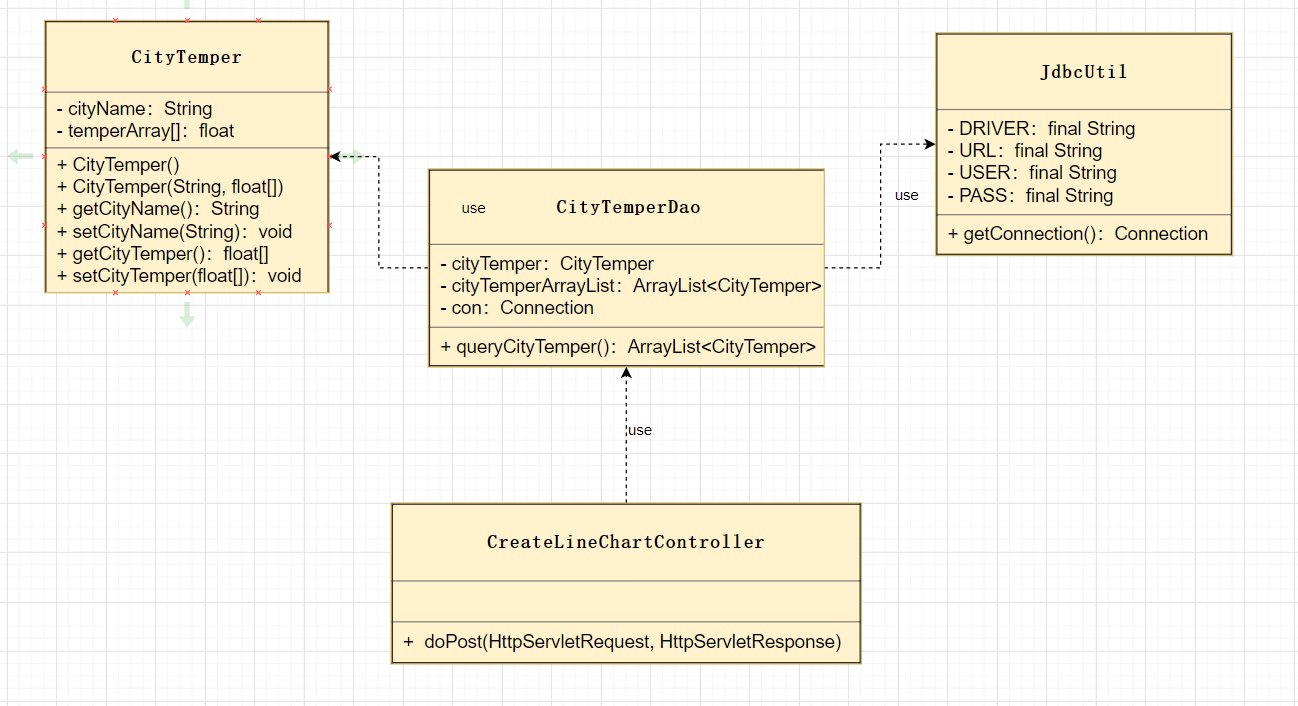


图2-2-2 功能2类图

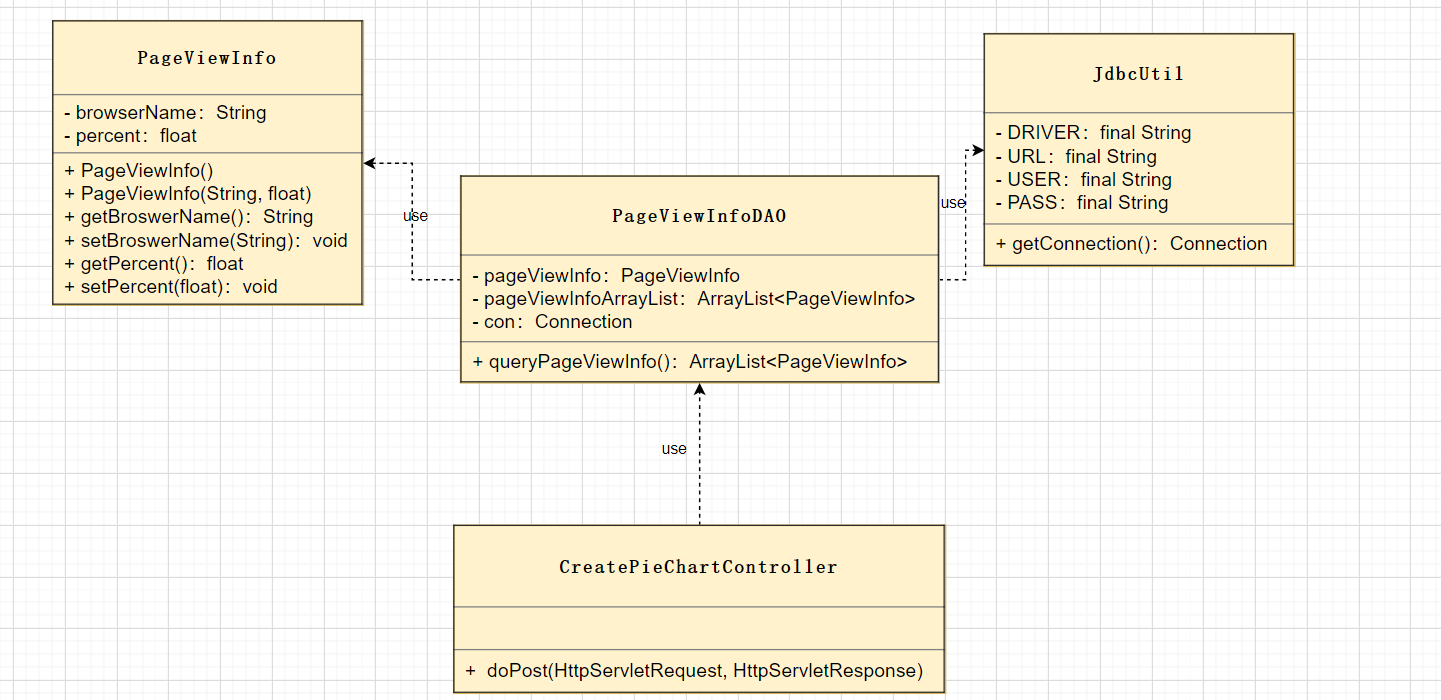
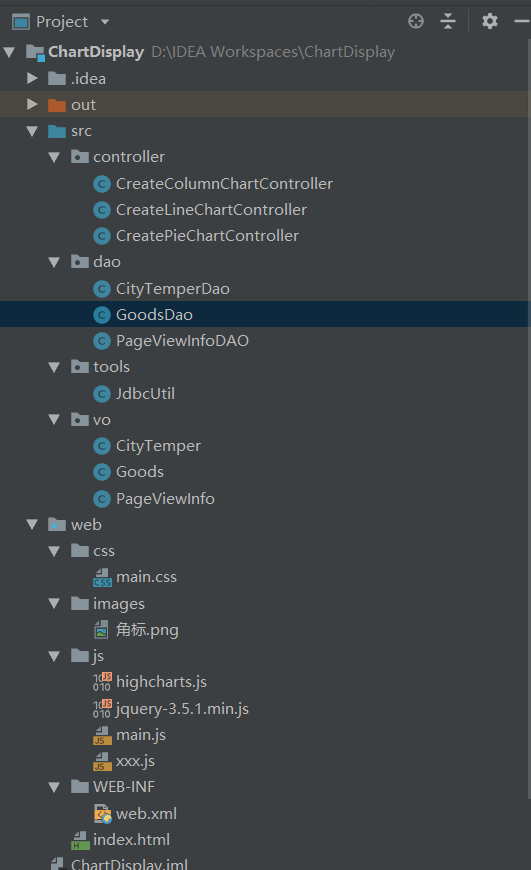


图2-2-3 功能3类图

# **3 系统实现**

## 3.1 项目结构



## 3.2 配置文件

web.xml文件：

<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"  
 version="4.0">  
</web-app>

## 3.3 VO类

**3.3.1 Price.java**

商品信息（商品、销售额）类

package vo;

public class Price {

private String merchandise; //商品

private float salesAmount; //销售额

public Price() {

}

public Price(String merchandise, float salesAmount) {

this.merchandise = merchandise;

this.salesAmount = salesAmount;

}

public String getGoodsName() {

return merchandise;

}

public void setGoodsName(String merchandise) {

this.merchandise = merchandise;

}

public float getsalesAmount() {

return salesAmount;

}

public void setsalesAmount(float salesAmount) {

this.salesAmount = salesAmount;

}

}

**3.3.2 Number.java**

城市气温类（城市名、温度字符串）

package vo;

public class Number {

private String merchandise;//商品

private float numberArray[];//商品数量

public Number() {

}

public Number(String merchandise, float[] number) {

this.merchandise = merchandise;

this.numberArray = number;

}

public String getmerchandise() {

return merchandise;

}

public void setmerchandise(String merchandise) {

this.merchandise = merchandise;

}

public float[] getNumber() {

return numberArray;

}

public void setNumber(float[] number) {

this.numberArray = number;

}

}

**3.3.3 BrowseInfo.java**

浏览商品量占比

package vo;

public class PageViewInfo {

private String browserName;//浏览器名字

private float percent;//浏览商品占比

public PageViewInfo() {

}

public PageViewInfo(String broswerName, float percent) {

this.browserName = broswerName;

this.percent = percent;

}

public String getBroswerName() {

return browserName;

}

public void setBroswerName(String broswerName) {

this.browserName = broswerName;

}

public float getPercent() {

return percent;

}

public void setPercent(float percent) {

this.percent = percent;

}

}

## 3.4 DAO类

### **3.4.1 PriceDao.java**

从数据库获取商品信息，并按照商品的销售额从大到小排列在list中。

package dao;

import tools.JdbcUtil;

import vo.Price;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

public class PriceDao {

public ArrayList<Price> queryPrice() {

Price price = null;

ArrayList<Price> priceArrayList = new ArrayList<>();

Connection con = JdbcUtil.getConnection();

String sql = "select \* from t\_goods";

PreparedStatement ps = null;

try {

ps = con.prepareStatement(sql);

} catch (SQLException e) {

e.printStackTrace();

}

ResultSet rs = null;

try {

rs = ps.executeQuery();

} catch (SQLException e) {

e.printStackTrace();

}

try {

while (rs.next()) {

String merchandise = rs.getString("merchandise");

float salesAmount = (float) rs.getInt("salesAmount") / 10000;

price = new Price(merchandise, salesAmount);

priceArrayList.add(price);

}

priceArrayList = sortBySalesAmount(priceArrayList);

} catch (SQLException e) {

e.printStackTrace();

}

JdbcUtil.closeAll(rs, ps, con);

return priceArrayList;

}

public ArrayList<Price> sortBySalesAmount(ArrayList<Price> list) {

for (int i = 0; i < list.size() - 1; i++) { //0~n-1

for (int j = 0; j < list.size() - 1 - i; j++) {

if (list.get(j + 1).getsalesAmount() > list.get(j).getsalesAmount()) {

Price number = list.get(j);

list.set(j,list.get(j+1));

list.set(j+1,number);

}

}

}

return list;

}

}

**3.4.2 NumberDao.java**

从数据库获取商品与数量的关系

package dao;

import tools.JdbcUtil;

import vo.Number;

import java.sql.\*;

import java.util.ArrayList;

public class NumberDao{

public ArrayList<Number> queryNumber(){

Number number = null;

ArrayList<Number> numberArrayList = new ArrayList<>();

Connection con = JdbcUtil.getConnection();

String sql = "select \* from t\_Number";

PreparedStatement ps = null;

try {

ps = con.prepareStatement(sql);

} catch (SQLException e) {

e.printStackTrace();

}

ResultSet rs = null;

try {

rs = ps.executeQuery();

} catch (SQLException e) {

e.printStackTrace();

}

try {

while (rs.next()) {

String foodName = rs.getString("foodName");

float[] numberArray = new float[12];

number = new Number(foodName,numberArray);

numberArrayList.add(number);

}

} catch (SQLException e) {

e.printStackTrace();

}

JdbcUtil.closeAll(rs, ps, con);

return numberArrayList;

}

}

**3.4.3 BrowseInfoDAO.java**

从数据库获取浏览量信息

package dao;

import tools.JdbcUtil;

import vo.BrowseInfo;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

public class BrowseInfoDAO {

public ArrayList<BrowseInfo> queryBrowseInfo() {

BrowseInfo browseInfo = null;

ArrayList<BrowseInfo> pageViewInfoArrayList = new ArrayList<>();

Connection con = JdbcUtil.getConnection();

String sql = "select \* from t\_pageview\_info";

PreparedStatement ps = null;

try {

ps = con.prepareStatement(sql);

} catch (SQLException e) {

e.printStackTrace();

}

ResultSet rs = null;

try {

rs = ps.executeQuery();

} catch (SQLException e) {

e.printStackTrace();

}

try {

while (rs.next()) {

String browserName = rs.getString("browserName");

float percent = rs.getFloat("percent");

browseInfo = new BrowseInfo(browserName, percent);

pageViewInfoArrayList.add(browseInfo);

}

} catch (SQLException e) {

e.printStackTrace();

}

JdbcUtil.closeAll(rs, ps, con);

return browseInfoArrayList;

}

}

## 3.6 工具包 JdbcUtil.java

连接数据库

package tools;

import java.sql.\*;

public class JdbcUtil {

private static final String DRIVER = "com.mysql.jdbc.Driver";

private static final String URL = "jdbc:mysql://localhost:3306/highcharts";

private static final String USER = "root";

public static final String PASS = "123456";

public static Connection getConnection() {

try {

Class.forName(DRIVER);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

Connection con = null;

try {

con = DriverManager.getConnection(URL, USER, PASS);

} catch (SQLException e) {

e.printStackTrace();

}

return con;

}

public static void closeAll(ResultSet rs, PreparedStatement pst ,Connection con) {

if(rs != null) {

try {

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(pst != null) {

try {

pst.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(con != null) {

try {

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

## 3.7 服务层

**3.7.1 CreateColumnChartController**

@WebServlet(urlPatterns = "/creatColumnChart.do")

public class CreateColumnChartController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String jsonStr;

PriceDao priceDao = new PriceDao();

ArrayList<Price> goodsArrayList = priceDao.queryPrice();

jsonStr = new Gson().toJson(priceArrayList);

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

System.out.print("1.条形图信息：");

System.out.println(jsonStr);

System.out.println();

out.print(jsonStr);

out.flush();

out.close();

}

}

**3.7.2 CreateLineChartController**

@WebServlet(urlPatterns = "/createLineChart.do")

public class CreateLineChartController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String jsonStr;

NumberDao numberDao = new NumberDao();

ArrayList<Number> cityTemperArrayList = numberDao.queryNumber();

jsonStr = new Gson().toJson(numberArrayList);

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

System.out.print("2.折线图信息：");

System.out.println(jsonStr);

System.out.println();

out.print(jsonStr);

out.flush();

out.close();

}

}

**3.7.3 CreatePieChartController**

@WebServlet(urlPatterns = "/createPieChart.do")

public class CreatePieChartController extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String jsonStr;

BrowseInfoDao browseInfoDao = new BrowseInfoDao();

ArrayList<BrowseInfo> browseInfoArrayList = browseInfoDao.queryBrowseInfo();

//转成json

jsonStr = new Gson().toJson(BrowseInfoArrayList);

response.setContentType("text/html;charset=utf-8");

PrintWriter out = response.getWriter();

System.out.print("3.饼图信息：");

System.out.println(jsonStr);

System.out.println();

out.print(jsonStr);

out.flush();

out.close();

}

}

## 3.8 用户界面

### **3.8.1 index.html**

网页文件

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>lqy的highcharts</title>

<script src="js/jquery-3.5.1.min.js"></script>

<script src="js/highcharts.js"></script>

<script src="js/main.js"></script>

<link href="css/main.css" rel="stylesheet" type="text/css"/>

</head>

<body>

<!-- 图表容器 DOM -->

<div class="div1">

<img src="images/2030966.jpg" alt="Study Website" width="170px" >

</div>

<div><font size="8px" color="greenyellow">Highcharts</font></div>

<br /><div class="line1"/><br />

<div><button class="updata" onclick="createColumnChart()">条形图</button><br/></div>

<div id="container1" style="min-width:800px;height:1000px;"></div>

<br /><div class="line1"/><br />

<div><button class="updata" onclick="createLineChart()">折线图</button><br/></div>

<div id="container2" style="min-width:800px;height:1000px;"></div>

<br /><div class="line1"/><br />

<div><button class="updata" onclick="createPieChart()">饼图</button><br/></div>

<div id="container3" style=min-width:800px;height:1000px;"></div>

</body>

</html>

**3.8.2 main.css**

网页的样式文件

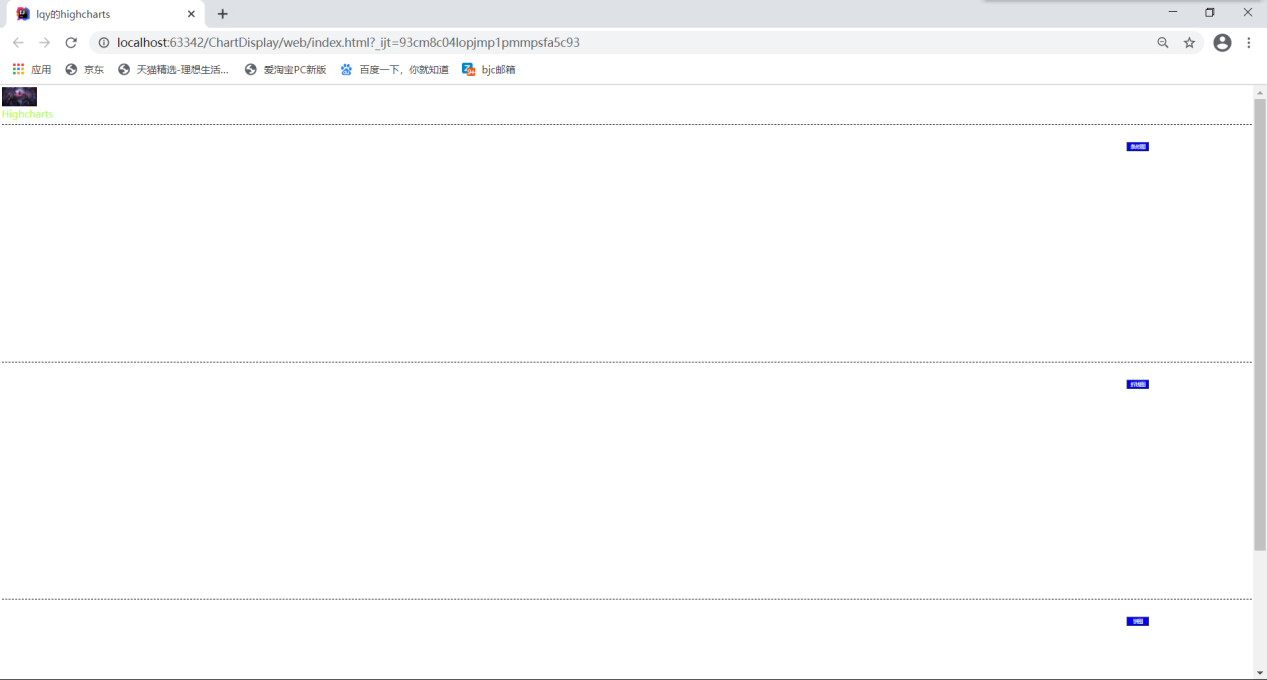
.line1{  
 width:100%;  
 border-top:6px black dashed;  
}  
  
.line2{  
 width:100%;  
 border-top:2px black dashed;  
}  
  
.div1  
{  
 top: 5%;  
 left: 10%;  
}  
  
.updata{  
 width:110px;  
 height:45px;  
 background-color:blue;  
 color: white;font-size: 24px;  
 cursor: pointer;  
 margin-left: 90%;  
 margin-top: 1%;  
}

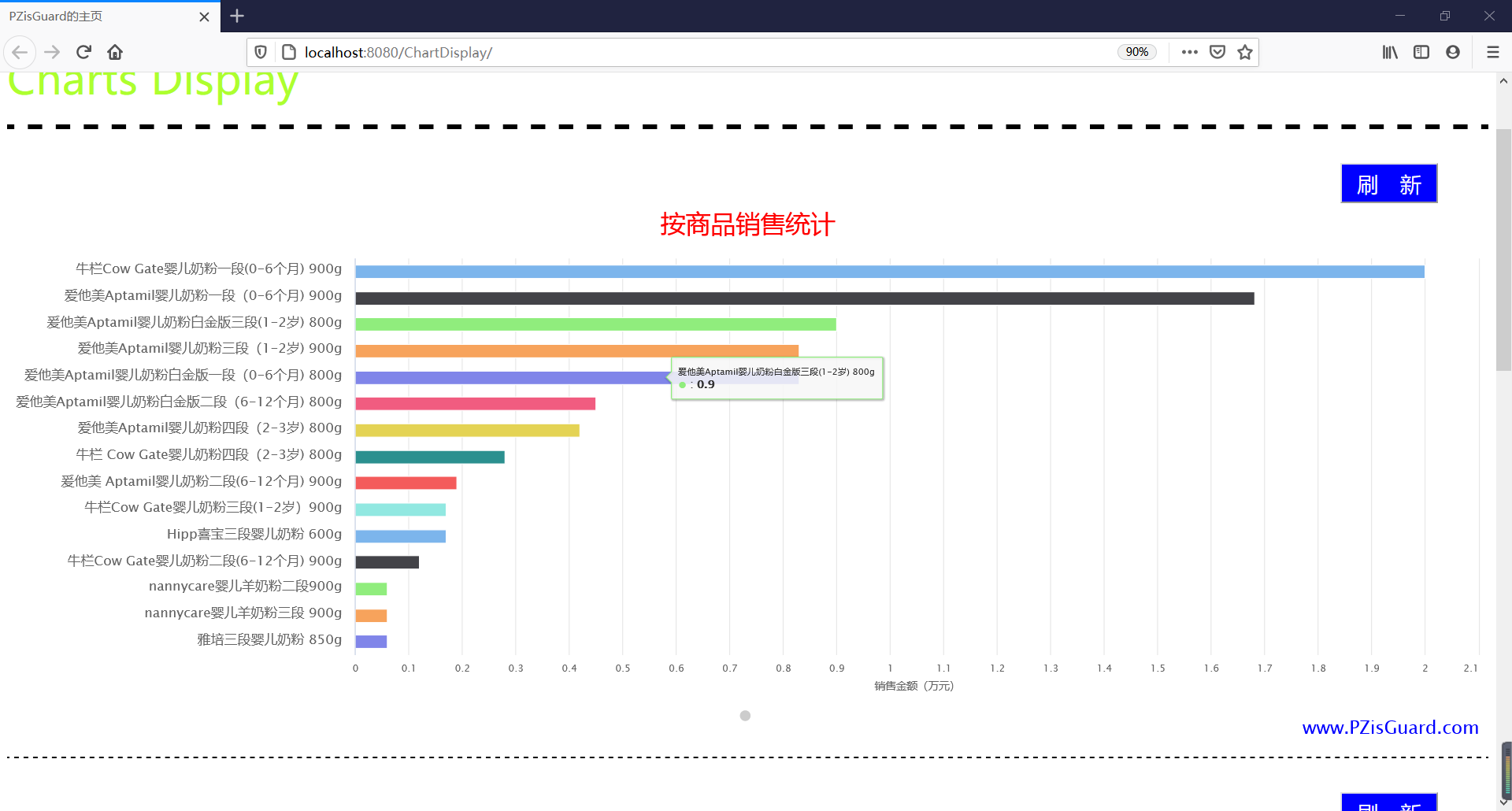
**3.8.3 jquery-3.5.1.min.js**

**3.8.4 highchart.js**

# **4 系统测试**

**4.1 生成、刷新条形图(按商品数量从大到小排列)**

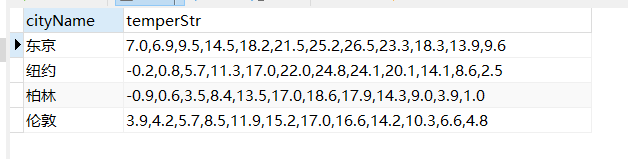


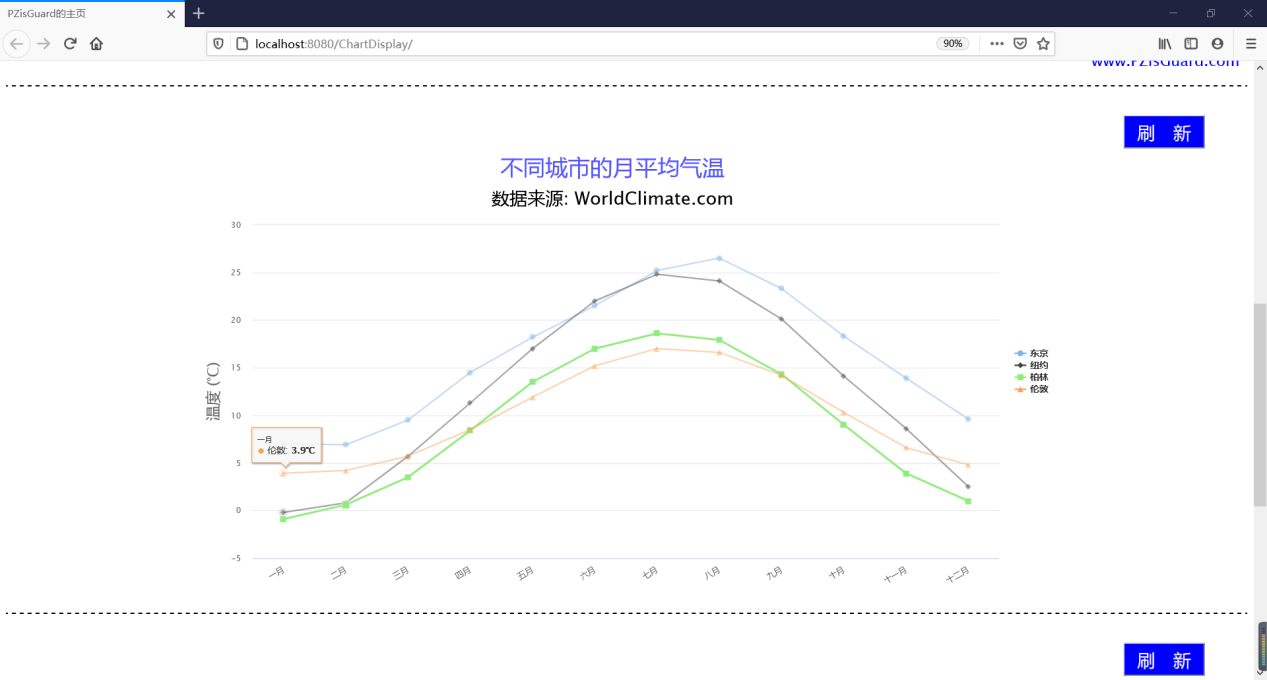


更新数据库后、点击“刷新”按钮：



**4.2 生成、刷新折线图**

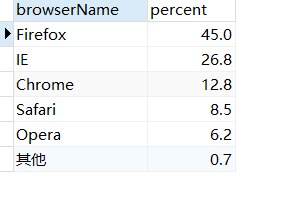


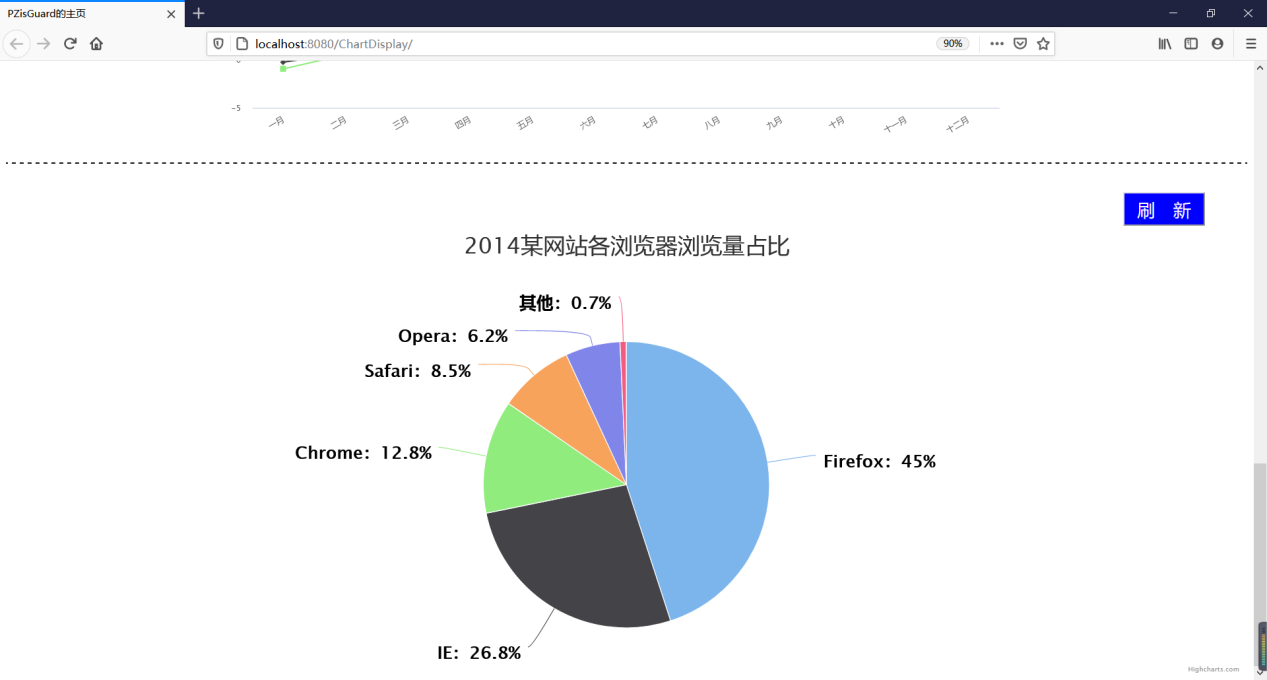


更新数据库后、点击“刷新”按钮：

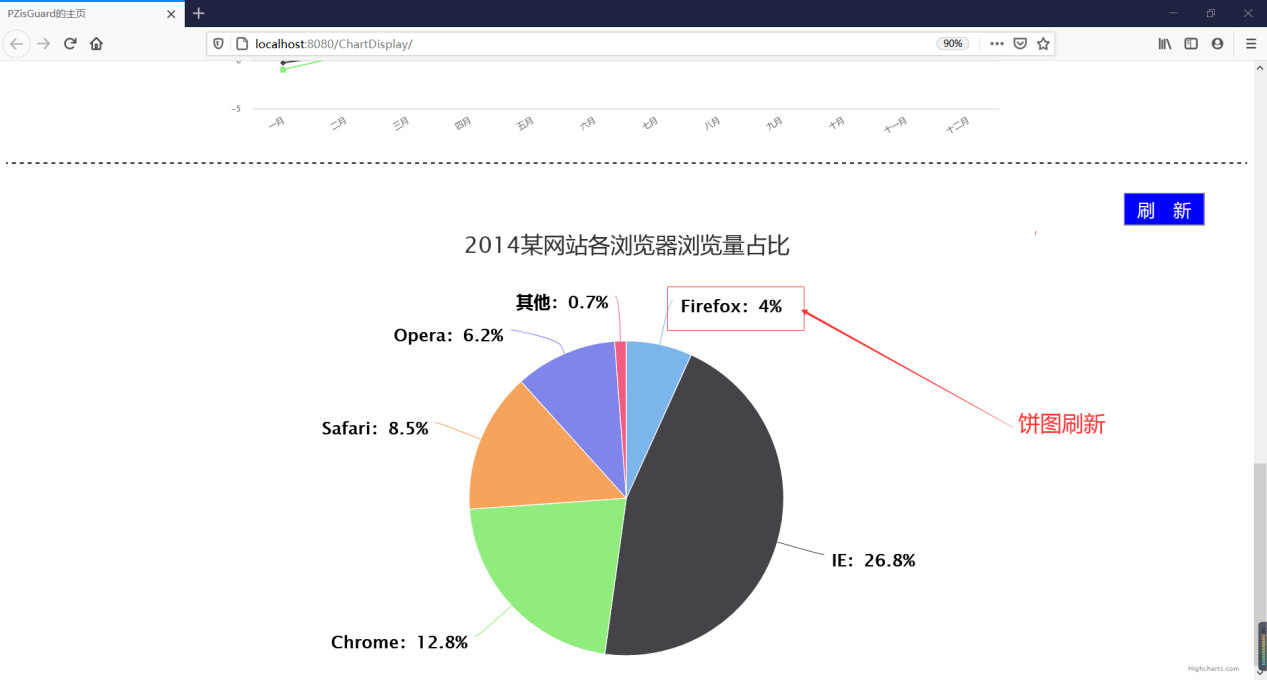


**4.3 生成、刷新饼图**





更新数据库后、点击“刷新”按钮：



# **5 系统总结**

HighCharts界面美观，由于使用JavaScript编写，所以不需要像Flash和Java那样需要插件才可以运行，而且运行速度快。另外HighCharts还有很好的兼容性，能够完美支持当前大多数浏览器。主要问题为后台数据库的处理和分析，数据库的使用不是很熟，很快就能利用组件来实现多种图状，但是要调用数据库比较麻烦，因此写了小样来测试。每次更新数据都需要自己去提交，还是很麻烦，要是能够自己更新数据库信息就好了。

课程收获：web学习不是很好，要在寒假好好补一补java和前端的知识，web是一个很有用的实战技术。

课程建议：老师对学生严苛是很好的，希望老师继续对我们严厉。

发展方向：硬件工程师

自评成绩：*80*